Handling Emergencies in Autonomous Systems with An Episode-Incident-Alert Workflow

Paul Baker, Kai-dee Chu, and Cindy Starr
Global Science and Technology, Inc.

Julie Breed
NASA-Goddard Space Flight Center

Jeffrey Fox
Pacific Northwest National Laboratory

Mick Baitinger
NEXTGEN Solutions, Inc.

Paul Baker ❊ pbaker@gsti.com ❊ http://abita.gsti.com

What Does Workflow Have To Do With Autonomy?

Autonomous operations are inevitable - computer systems will:

- **perform routine steps**
- **diagnosis problems and correct many of them**
- **assess conditions and modify plans accordingly**
- **save lots of money by reducing staffing so that more funds are available for scientific and industrial applications.**
- **And, they will sometimes find a problem only an engineer can correct!**

This paper is not concerned with how to achieve autonomy.

Workflow processing is concerned with organizing an effective response to a problem.

The setting for our discussion is an Emergency Response System or ERS that was developed to organize responses to unplanned situations that occur during satellite operations. We discuss the ERS first.

The new work we want to introduce is a model for interpreting the telemetry stream so that the ERS receives better information and people respond better in an emergency.

Paul Baker ❊ pbaker@gsti.com ❊ http://abita.gsti.com

The Major Concern: Surprise!!

A low-cost ground system operation must succeed with little attention from operators or engineers.

We anticipate that all future manual interventions will occur in unplanned, emergency situations.

We should be concerned for the performance of people thrown into an unfamiliar emergency situation.

Every request for attention from the autonomous system will be an unexpected surprise for the staff. When the operation is threatened, the surprise becomes an emergency!

We expect engineers and part-time operators to drop whatever they are doing, jump on a problem, and correct it. How can we be sure they will respond quickly? effectively? safely?

Information engineering in the past only needed to deal with the routine.

Information engineering for the future must deal with the unexpected.

Paul Baker ❊ pbaker@gsti.com ❊ http://abita.gsti.com

What is Workflow?

Workflow is a principle of software design that tasks the system with the completion of certain objectives. The system either does the work itself or cajoles people into doing their share.

We use a combination of E-Mail and a Paging Service to alert people to what is needed.

- **The workflow system features a backup to these alerts. If a person does not respond to an alert in the expected time, another alert can be issued.**

Workflow involves the flow of documents and artifacts. At least some of the documents must exist in the workflow system's data base.

- **We are using Lotus Notes, which has a built-in data base.**
- **Our environment keeps information in files and Web servers. Notes documents can incorporate references to this type of information.**

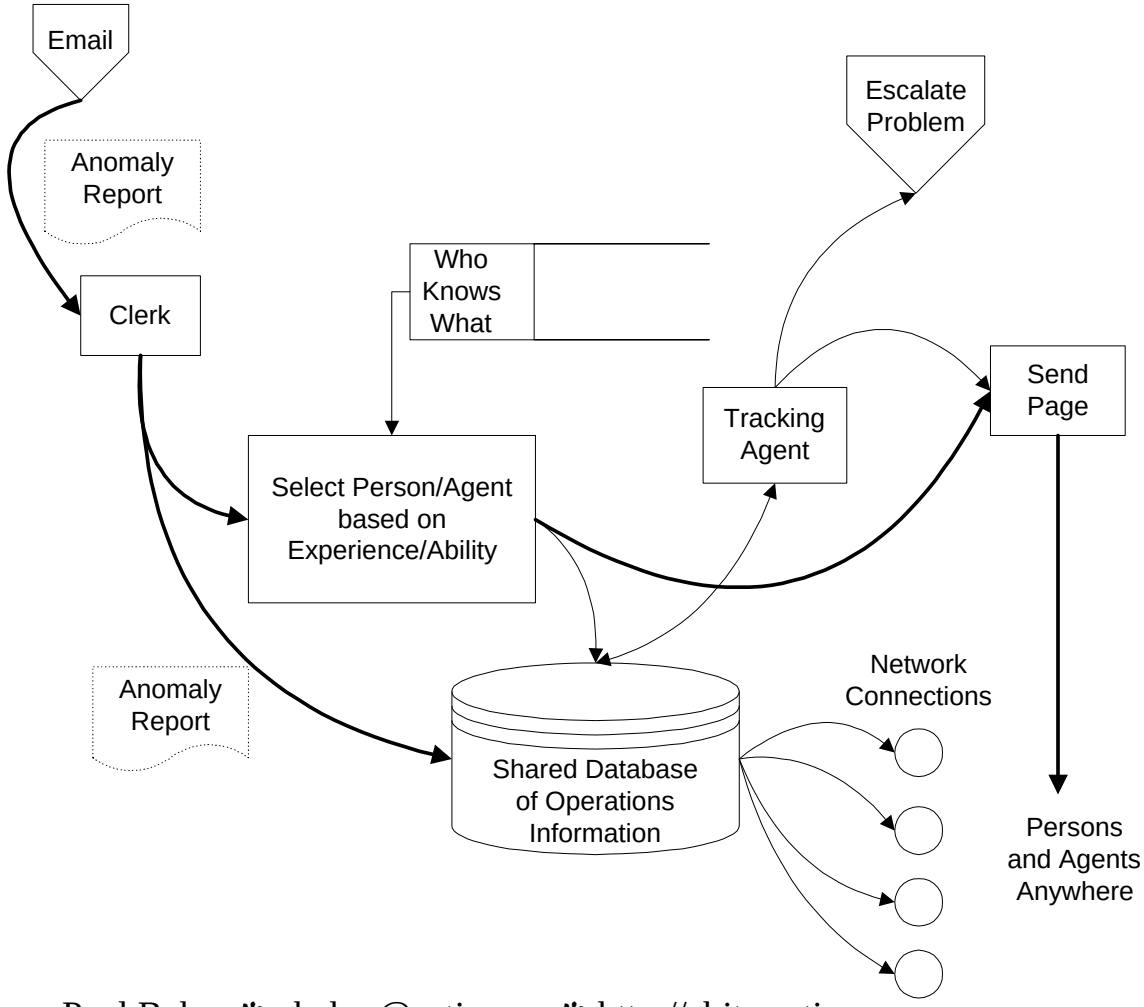
Workflow system incorporate "agents" that keep the workflow operation moving even if people are diverted or unavailable.

- **In Lotus Notes, agents respond to the arrival of e-mail, the arrival of documents and modifications to existing documents. Some agents run on a schedule.**

Paul Baker ❊ pbaker@gsti.com ❊ http://abita.gsti.com

Sample Workflow

E-mail activates a response, workflow system organizes and checks the response.



Paul Baker ❊ pbaker@gsti.com ❊ http://abita.gsti.com

Immediate Workflow Objectives

Anomalies are routed to the right person.

- **An agent acts as a clerk and compares E-Mail reports to a data base of capabilities then selects who or what should respond.**
- **The report is saved in a shared area accessible to authorized personnel using networked computing.**
- **Alert goes to a person via a pager.**
- **People can log in anywhere and investigate the problem.**

Agents make certain that there is progress on a problem.

- **A tracking agent checks for a timely response, sends the page again if no response, and escalates the problem if delay becomes unreasonable.**
- **Agents collect many reports for statistical purposes and route them appropriately for analysis.**

Paul Baker ❊ pbaker@gsti.com ❊ http://abita.gsti.com

Future Workflow Objectives

The same workflow can select the proper automatic agent, e.g. an expert system, to handle a problem within its scope of ability. The problem can be escalated to a human if the agent fails.

A more complex workflow can assemble material for "learning machines", e.g. case-based reasons so that the autonomous operation gets smarter.

Workflow procedures can remind staff of occasional but necessary operational duties.

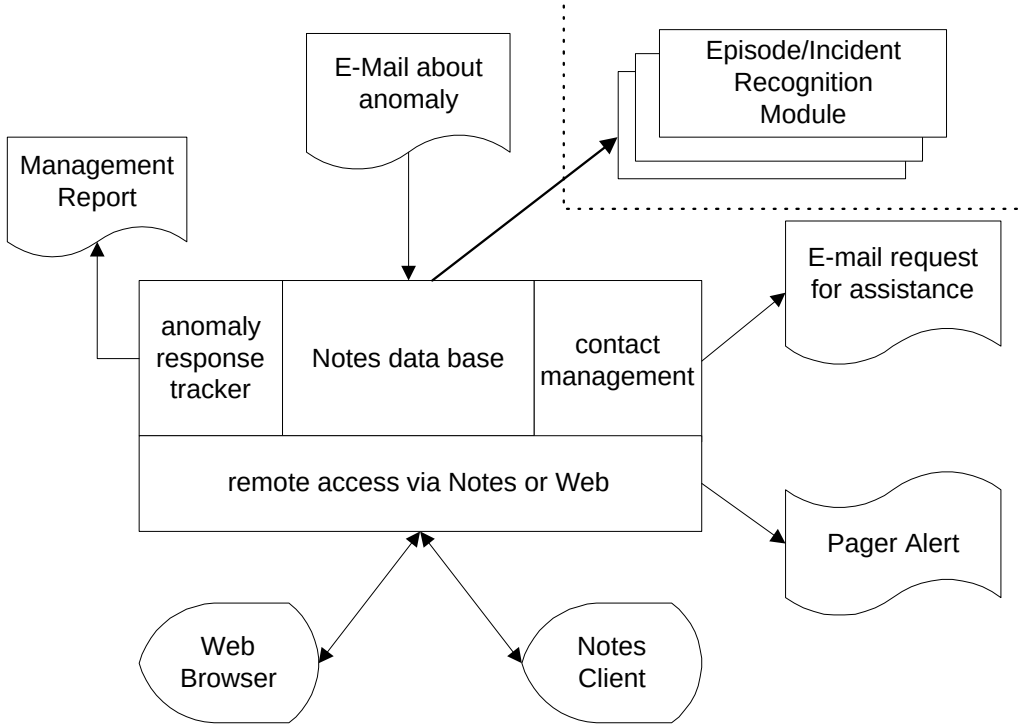
Better emergency response procedures to help people fix problems before damage is done. These will be added with a new subsystem for handling episodes - as described in the following slides.

Paul Baker ❊ pbaker@gsti.com ❊ http://abita.gsti.com

Current ERS Design

Currently, the ERS is built around a Lotus Notes data base including a Pager Gateway (Skytel) and a Web Server (Lotus Domino).

The E*I*A module is a new addition - and the main subject today.



Paul Baker ❊ pbaker@gsti.com ❊ http://abita.gsti.com

The E*I*A Model

The E*I*A Model is not a system, rather it is a concept for using a system to provide better operations. At the very least, the model helps classify information so that we can act on it effectively.

The model describes how to examine a telemetry stream, how to identify problems, and how to set the stage for problem resolution.

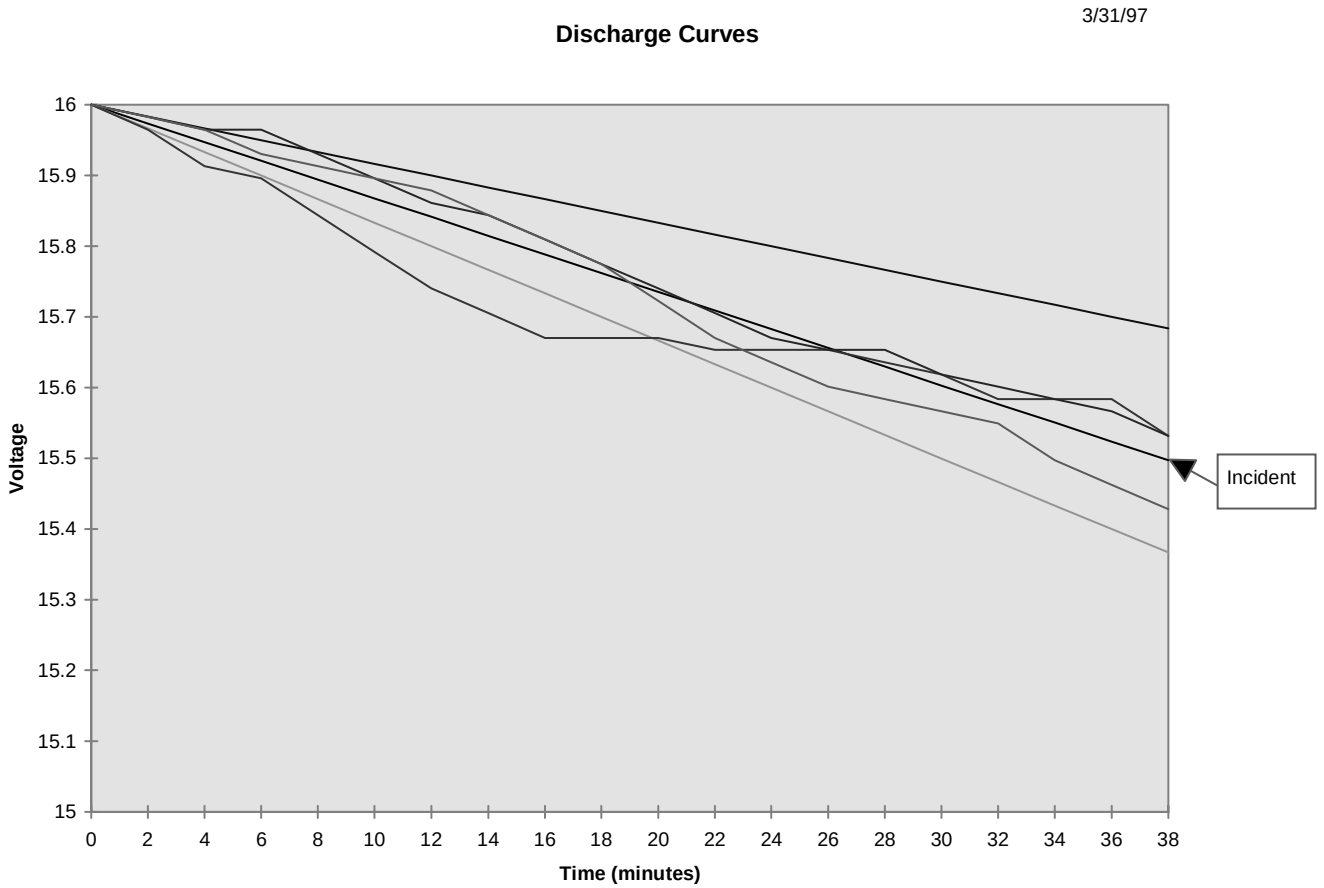
Components of the Episode-Incident-Alert or E*I*A model:

- **Episode: An *Episode* is any time sequence that deserves examination.**
  - **Episodes are recognized automatically; then, telemetry data are assembled to describe what happened during the episode.**
- **Incident: An *Incident* is an *Episode* that demands attention.**
  - **Automatic analysis programs examine telemetry data sets from each episode using various techniques to characterize the engineering state of the spacecraft.**
  - **Based on that analysis, an episode may be set aside or elevated to the status of an Incident.**
  - **The data set for an incident carries with it the telemetry data from the episode as well as any numeric output from the engineering analysis.**
- **Alert: An *Alert* is a notification of an *Incident* that is sent to an engineer or a specialized autonomous agent. Usually, an Alert contains a short explanation of the Incident. Details are obtained from the Incident data set.**

Paul Baker ❊ pbaker@gsti.com ❊ http://abita.gsti.com

Engineering Example

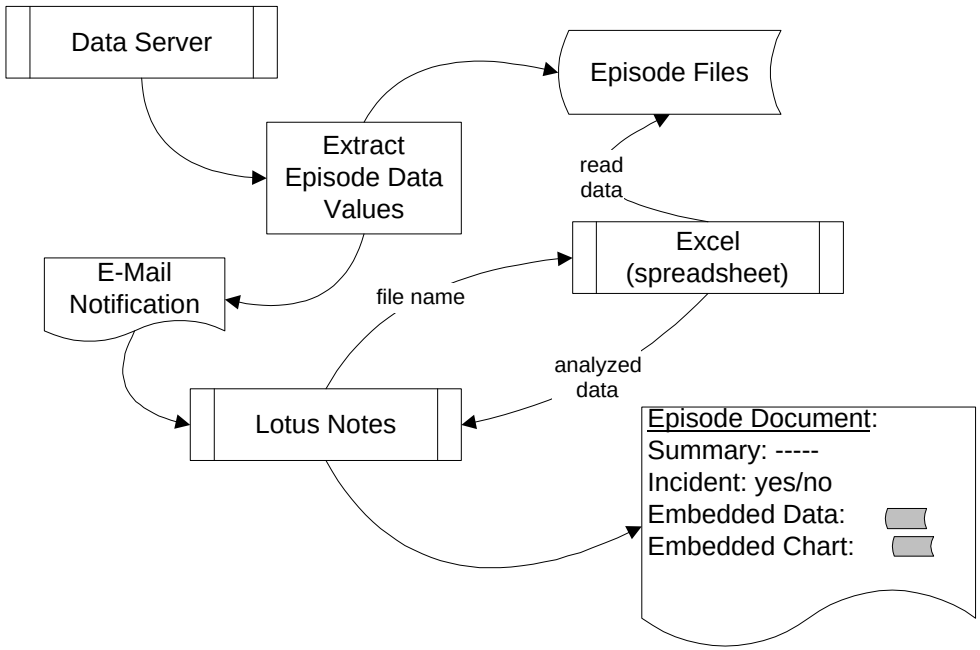
Here is an example of a series of episodes - 5 normal and 1 anomalous. Only engineering analysis can pick out the problem.



Paul Baker ❊ pbaker@gsti.com ❊ http://abita.gsti.com

Handling Engineering Episodes

The preceding example of "battery discharge curves", can be analyzed using a spreadsheet containing engineering formulae and calibration parameters established via preflight testing and trend analysis. The spreadsheet can automatically classify episodes for the ERS. The following data flow diagram explains how:



Paul Baker ❊ pbaker@gsti.com ❊ http://abita.gsti.com

Why Use the E*I*A Model and Workflow

Workflow procedures automatically

- **assemble the evidence, the episode**
- **run the engineering analysis to detect the incident**
- **create explanatory data: graphs, tables, numeric results, etc.**
- **Assemble all available information for the person who responds to the Alert.**

The information package helps provide an engineer with the proper context to understand the problem and solve it. It reduces the element of surprise and reduces the risk that the human will respond poorly when thrown suddenly into the middle of a satellite operations problem.

As the preceding example shows, the idea also provides a way to have ongoing engineering analysis applied routinely without compelling an engineer to perform tedious, repetitive work in the operations center. Morever, the routine analysis will supply better information for trend analysis.

Paul Baker ❊ pbaker@gsti.com ❊ http://abita.gsti.com

Handling Event Messages

Event messages are sent from the ground system and may contain warnings or alerts that need to be handled by an engineer or operator. But there are a couple of problems:

- **Messages are all jumbled together. One message may indicate a problem but all the messages around it are unrelated. An engineer who is just arriving to fix an emergency situation must first sort out the messages.**
- **One problem may trigger an avalanche of messages, each of which is worth a pager alert to an engineer. The paging service - not to mention the engineer - would be overwhelmed by many individual alerts. Too much time would then be spent simply clearing the queue.**

The E*I*A model is applied to correct the second problem by defining episodes that cover a fixed period of time. Alerts are sent for an episode, not an individual event. The length of an episode is adjusted so that:

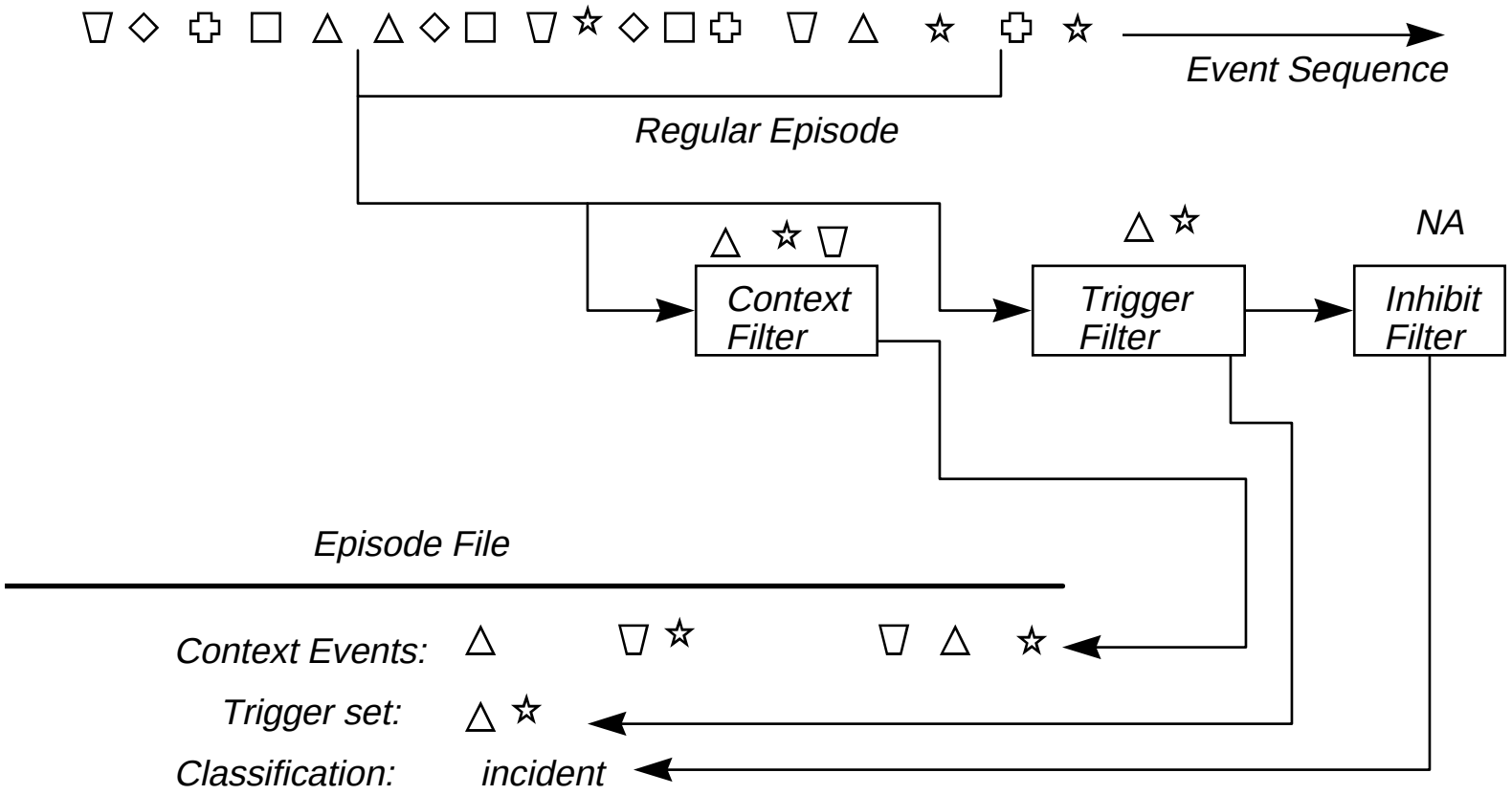
- **We want to make the episodes long in order to restrict the number of alerts.**
- **A shorter length results in a faster response because an alert is not generated until the episode is complete.**

Note, this topic is not covered in the conference publication. If interested, see the longer version of the paper on the web site : http://abita.gsti.com/eia.htm

Paul Baker ❊ pbaker@gsti.com ❊ http://abita.gsti.com

Sorting Out the Messages

Illustration of a single episode filter pipeline:



Paul Baker ❋ pbaker@gsti.com ❋ http://abita.gsti.com

Event Filter Steps

Each filter pipeline has the goal of creating a coherent information package for an engineer involving one subject. There may be many pipelines because there are many technical subjects to consider.

The three filters in each pipeline have the following functions:

Context Filter

- **Selects only those events that are relevant to a subject**
- **Casts a wide net and catches evidence for problems as well as evidence about related situations.**

Incident Filter or Trigger Filter

- **Selects events that indicate there is a problem. If there is such an event, and the event passes the Inhibit Filter, the episode will be classified as an incident and it will produce an alert.**

Inhibit Filter

- **Selectively blocks events that pass the Incident Filter.**
- **Intended for temporary use - to block alerts based on certain messages that the engineering team has seen before and doesn't want to see now.**
- **The aim here is to avoid modifying the broader incident filter when it is necessary to disable certain messages.**

Paul Baker ❊ pbaker@gsti.com ❊ http://abita.gsti.com

Summary

Future autonomous satellite operations will sometimes need help. The automatic system needs to be complemented with systems that aid people work effectively on a sudden, emergency problem.

The Emergency Response System (ERS) is a prototype using Lotus Notes, E-Mail, Web, and Paging Service to accomplish this overall objective.

The E*I*A Model is a design concept that helps build workflow applications that run in the ERS.

- **The ERS runs many simultaneous applications because each one handles a restricted subject and more may be added over the lifetime of the mission.**
- **We have implemented an E*I*A application that responds to Engineering Episodes by running spreadsheet models of the hardware automatically. This application brings engineering expertise closer to everyday operation.**
- **We have implemented an E*I*A application that creates summaries of related system events. This type of application prevents abuse of the paging service when many related anomalous events occur.**
- **The system will be introduced first for the TRACE mission with others to follow.**

Paul Baker ❊ pbaker@gsti.com ❊ http://abita.gsti.com